Location Assisted Vertical Handover Algorithm for QoS Optimization in End-to-End Connections

Dam, Martin S.; Christensen, Steffen R.; Mikkelsen, Lars Møller; Olsen, Rasmus Løvenstein

Published in:
Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on

DOI (link to publication from Publisher):
10.1109/TrustCom.2012.197

Publication date:
2012

Document Version
Early version, also known as pre-print

Location Assisted Vertical Handover Algorithm for QoS Optimization in End-to-End Connections

Authors
Martin S. Dam, ms08@student.aau.dk
Steffen R. Christensen, srch08@student.aau.dk
Lars M. Mikkelsen, lmikke08@student.aau.dk
Rasmus L. Olesen, rlo@es.aau.dk

Aalborg University
Network and Distributed Systems
Fredrik bajers vej 7
9220 Aalborg East
Denmark

***Abstract*—When devices are mobile, they potentially move within range of several different wireless networks, which are not utilized due to security, inactive radios or it is just plain cumbersome for the user to exploit them. This paper proposes a vertical handover algorithm to utilize these networks, which selects the best available network at a given time, factoring in prediction of user movement, energy consumption, QoS of connections and application demand for the end-to-end connection in a (multiple) client/server setting. The algorithm allows support for different mechanisms of interaction, where we in this paper focus on 1) peer-to-peer in a WLAN setting, 2) p2p behind NAT and 3) what we call a server bounce mechanism. The algorithm is supported by a User-specific Virtual Network to obtain required network state information. Experimental tests are conducted, using both simulations and actual implementation on Android based tablets. The simulations cover a wide range of scenarios for two mobile users in an urban area with ubiquitous cellular coverage, and shows our algorithm leads to increased throughput, with fewer handovers, when considering the end-to-end connection than to other handover schemes. The implementation utilizing real world networks, shows the feasibility of the algorithm due to its low complexity compared to other algorithms.**

I. Introduction

The extensive WiFi coverage in urban environment provides the possibility to improve mobile devices' connectivity compared to e.g. cellular networks, with respect to Quality of Service (QoS) parameters as throughput, delay, jitter and network restrictions. Furthermore, as WiFi consumes less energy than 3G [1], it is clear, that a potential improvement of performance and energy consumption can be achieved by choosing the point of attachment (POA) that provides the best QoS at any given time.

However, in the physical world a handover between POAs will result in a downtime associated with the change of wireless interface due to access point (AP) association, DHCP look-up and IP (re)configuration, which takes time. In addition, a vertical handover (VHO) may lead to further energy consumption due to the added communication. Finally, sudden changes of the IP address may corrupt existing service sessions. Therefore, the amounts of VHOs should be kept low, by making informed and intelligent decisions of when and how to trigger a VHO.

A. Background & Motivation

Other handover algorithms, e.g. as proposed by [2], [3] and [4], consider a single device and its connection to either a base station (BS) or an access point (AP), and not the applications end-to-end connections. Leaving the end-to-end aspect out of the network handover consideration, can be inefficient. For example: If device A is streaming data to device B, it might not be beneficial for device A to make a VHO if there is a congestion at device B. In typical handover algorithms, device A may then change network to the AP, which results in a downtime of the stream. Other VHO algorithm, also does not consider the connection scheme (server/client, p2p etc), and suffers from not incorporating real world applications.

In this paper we propose an algorithm, that selects the best link considering the application end-to-end connection. The algorithm is inspired by Nash equilibrium [5] known from game theory. This theory can be applied when each participant is assumed to know the strategy of the other participants, and that each will have no gain by individually changing strategy, if the others keep their current strategies. However, a handover algorithm considering the end-to-end connection, will require a larger amount of information about the involved devices, such as movement, network capabilities, network state and available networks. Simplicity of the algorithm is therefore a key objective, and for that reason we will also show the feasibility of the algorithm later on by implementing it in a real setting.

B. System Overview, Assumptions & Constraints

A conceptual overview of the system setup is illustrated in Figure 1. The key entity involved in the handover process is the Dynamic Application Migration server (DAM server), which basically ensures the execution of the algorithm.

For the algorithm to work an underlying system which allows data synchronization, is assumed present along with a Cartesian network map, which provides updated information of the available POAs and their properties[1] in a specific geographical area. To ensure these two assumptions are met, we use the User-specific Virtual Network (UVN) from

[1]Network properties include throughput, delay and jitter

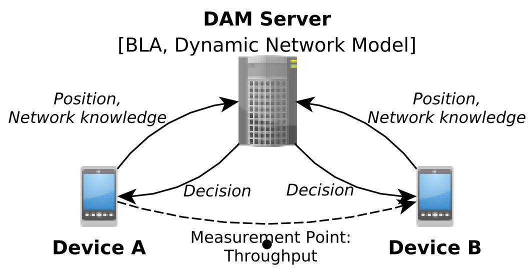

Figure 1: Simplified system overview for the considered problem. The server maintains a dynamic network model based on data gathered from the devices. The Best Link Algorithm (BLA, to be defined) is deployed on the server. The throughput in an end-to-end connection is measured, and is what should be improved.

the project Dynamic Application Migration, [6] at Aalborg University. The UVN data is located on the DAM server, and the resources are assumed to be available for the deployment of the algorithm. Furthermore, we assume that a seamless handover protocol exists, that allows a data stream to continue after a successful VHO, thereby making the VHO transparent to the application.

In this paper we limit ourselves to TCP based streams on WiFi and 3G networks, and the proposed solution will be evaluated based on two devices under emulated network and movement conditions, supported by computer simulations to cover a wider range of scenarios.

The rest of the paper is structured as follows: Section II introduces network concepts, the quality factors used in the project and describes related projects. In Section III and IV the general scenario is presented, leading to a general problem description, which is reduced to fit the specific problem. A solution is obtained to the specific problem, and an algorithm for selecting the best link, based on extended network knowledge and prediction is derived. Section V presents the setup for the tests, and the results are presented in Section VI. In Section VII the conclusion is done based on the test results, and in Section VIII the proposed future work is stated. Section IX describes the various models used for the simulations and emulations.

II. General Concepts & Related Work

In this section the general concepts used throughout the paper will be introduced, followed by previous work of interest to the subject.

A. Network Concepts

We consider in this paper two different concepts of connecting two devices; Server bounce and Peer-to-Peer (p2p), but is easily extended to others due to the generality of the proposed algorithm. In the following we describe the important details of the two selected concepts.

In the **server bounce** scheme the devices establish connections to a server, which then pairs the devices together. When one of the devices then transmits data to the other, the server reroutes or bounces the packets to the other device. This scheme is considered always to be available for the devices, but offers a somewhat limited transfer rate as the communication must pass through a server.

In the **p2p** scheme the two devices establishes a direct connection to each other. But as most private users and networks are behind NAT, due to the Internet Service Provider or the internal LAN setup, the p2p scheme is not always directly applicable. To establish a p2p connection between devices "hidden" behind NAT, the concept of TCP hole punching needs to be applied, as described by [7] and [8]. The p2p scheme is considered sporadic, but offering a higher transfer rate, while not putting as much load on the system as the server bounce. A special case of LAN p2p is also utilized, when the Tx/Rx can be assumed symmetric.

B. Quality Parameters

The method of selecting the best link is partially based on comparing quality parameters of the physical and data link layer of the involved links. These quality parameters comprises the general QoS parameters such as bit rate, delay, jitter and packet loss probability, expressed as a quality factor Q. This quality factor is rated better for connection types that put less load on the system (e.g. the DAM server), in order to ensure scalability, and the lower the quality factor the better QoS. The quality factor Q is an arbitrary function, and should be constructed to fit the actual implementation.

C. Previous & Related Work

The work done in [9] confirms that WiFi performs better than 3G networks in terms of QoS, even though 3G provides a more constant coverage than the sporadic coverage of WiFi networks, and concludes that WiFi suffers from long setup times, and in order to improve the utilization of the WiFi networks on mobile devices, this operation needs to be optimized.

As described by [3], the limited energy resources of mobile devices turns into a significant performance bottleneck, and the energy consumption will need to be considered in a VHO algorithm. The assumption is that 3G is more energy consuming than WiFi. This is supported by [1], which presents a measurement study of the energy consumption of 3G, GSM and WiFi. They conclude that 3G and GSM has an energy overhead, because they linger in a high power state even after the actual data transfer is complete, which is not the case when using WiFi.

In deciding when and if a device should do a VHO, some information regarding the movement of the device is needed. [2] proposes a location-based look-ahead handover prediction algorithm, which uses the handover time, location of APs and location prediction to decide whether a handover is profitable within a fixed time-window. A database containing locations of access points and their respective throughput is used to support the decision of the algorithm - this approach will be utilized and extended in this paper.

[2] concludes that an accurate location prediction is needed, due to the use of look-ahead windows, which makes the algorithm sensitive to noise. The complexity of the algorithm heavily depend on the look-ahead window size, and the algorithm is only feasible when at most 4 networks are within the look-ahead window, making it infeasible under normal circumstances. [2] also propose a Heuristic Look-ahead algorithm (HLA), with lower complexity than their optimal solution, but still computational heavy due to the look-ahead window. However, only a single device and the connection to AP or BS is considered, and not an end-to-end connection.

We will in this paper construct an algorithm that considers these issues, and show that we can obtain a better performance than a selected number of approaches.

III. General System Model

In this section we present a generalized system model and defines a formal description of the problem.

A. General Scenario

The possible options of which a connection scheme can take will be formulated as a directed graph $G = (V, E)$ for the general case, from transmitter s to receiver t, where each path and node in the graph represents a connection combination, see Figure 2. Each combination is composed of N connection schemes, in the shown case $N = 3$, with a) p2p via LAN (LANp2p), b) p2p through NAT using TCP hole punching (HPp2p), and c) server bounce (SB) and M number of interfaces (3G, WiFi, WiMAX, Bluetooth, etc.).

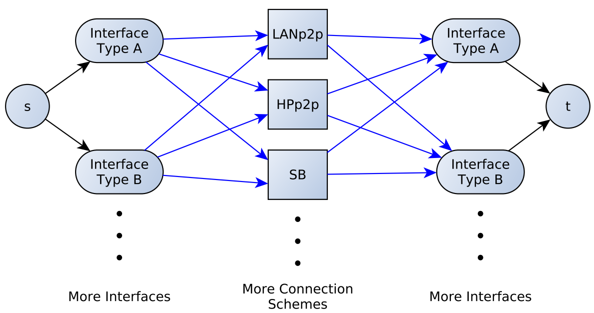

Figure 2: A general graph of the possible connection combinations. The connection points LANp2p, SB and HPp2p should be added according to the limitations of the specific interface type.

Each edge in the graph has a set of characteristics, which are defined as follows:

- A flow $f(u, v)$: The actual data flow in bit/s
- A capacity $C(u, v)$: Available bandwidth on the connection in bit/s.
- A cost $a(u, v)$: Describing the cost of using a given connection, e.g. in terms of energy or payment plan (3G), $a(u, v) \geq 0$.
- A state $b(u, v)$: A binary function, that indicates the state (on/off) of the link, $b(u, v) \in \{0, 1\}$.
- A quality factor $Q(u, v)$: An arbitrary quality function of the connection, where the function is a parameter for the algorithm. The worse the QoS the higher a value.
- A demand factor D:

B. General Formal Problem Description

Based on III-A a general cost function $g(u, v)$ for each edge is defined:

$$g(u,v) = \begin{cases} \left((1-D) \cdot \dfrac{a(u,v)}{f(u,v)} + D \cdot Q(u,v) \right) \cdot b(u,v) & \text{if } f(u,v) \neq 0 \\ a(u,v) \cdot b(u,v) & \text{if } f(u,v) = 0 \end{cases}$$

The rationale behind $g(u, v)$ is to select the link with the best utilization and offering the demanded QoS. If a connection is used ie. $b(u, v) = 1$, the cost of using this connection is applied via $a(u, v)$, but when more flow is sent through the connection the cost decreases, hence the better utilization (flow/energy ratio), the lower value of $g(u, v)$. The impact of the quality is included via the function $Q(u, v)$ that sums each element from s to t, where the smallest element in $Q(u, v)$ minimizes the quality impact on $g(u, v)$, hence the path with best QoS will be selected with respect to the cost $a(u, v)$. When $f(u, v) = 0$ only the energy cost is considered, since QoS is only relevant when data is flowing. The quality demand $D \in [0, 1]$ is defined by the application, which shift the weight between energy cost and quality. E.g. a file transfer where the flow is important, would use a D $<$ 0.5 whereas a videostream that require low latency would use D $>$ 0.5.

With the function $g(u, v)$ defined, the minimization problem, that minimizes the cost and maximizes the flow, can be expressed as:

$$g_{min} = \min \underset{\text{over } f(u,v) \wedge b(u,v)}{\left\{ \sum_{(u,v)\in E} g(u,v) \right\}} ,$$

under the following constraints:

1) $f(u, v) \leq C(u, v)$ — Capacity constrains
2) $f(u, v \mid b(u, v) = 0) = 0$ — State flow condition
3) $\sum_{w\in V} f(u, w) = \sum_{w\in V} f(w, u) \;\; \forall \;\; u \in V \setminus \{s, t\}$ — Flow conservation
4) $\sum_{w\in V} f(s, w) > f_{min} \geq 0$ — Flow from s $\rightarrow$ t
5) $b(u, v) = b(v, u) = \{1, 0\}$ — State symmetry
6) $\sum_{(u,v)\in E} b(u, v) \geq 1$ — Minimum one connection
7) $a(u, v) = a(v, u)$ — Cost symmetry

Solving the minimization problem, will result in selecting the link most profitable at a given time with a given available network, under the constraints and cost function defined. However since the POA changes over time, as users with mobile devices moves between different access points and network coverage, the above minimization problem is required to be solved each time the network changes. Furthermore, since multiple wireless network signals can occupy the same coverage areas the minimization problem shall be iterated.

C. Restrictions to algorithm

The cost function does not incorporate any hysteresis zone, since it is assumed to be placed centrally, where position and network data has already been filtered by the information provider system, before being processed by the algorithm. Also, since the handover is associated with a downtime, in which updated network and position data can not be delivered to the server, this naturally incorporates a hysteresis zone to avoid rapid subsequent handovers. However, this is not sufficient for all cases, therefore the algorithm must return the same decision in two sequential computations spaced Δt in time, in order to act on the decision.

IV. Algorithm for Solving Minimization Problem

As mentioned earlier, we have limited our work in this paper to the following three connection schemes; LANp2p, HPp2p and SB, and two connection types (3G and WiFi). By assuming that each device only has one interface available at a time, an additional constraint to the state function can be specified.

$$\sum_{(u,v)\in E} b(u,v) = 4 \qquad \text{Exactly one connection}$$

This constraint will result in exactly one connection from s to t, as a result of the flow conservation and flow from $s \to t$ constraints.

Proof: All path length from s to t is 4 due to the construction of G. Since each vertex except s and t has flow conservation, and the flow from s need to be positive and larger than zero, the only possible solution is to let the flow path go from s to t, and since this is 4 steps, the above constraint will result in only one connection ■

Further, by constraining the cost $a(u,v)$ and $Q(u,v)$ on specific edges

$$a(u,v) = 0 \qquad \forall\, u,v \in V \setminus \{s,t\} \wedge (u,v) \in E \tag{1}$$

$$Q(u,v) = 0 \qquad \forall\, u,v \in \{s,t\} \wedge (u,v) \in E \tag{2}$$

essentially allows the combination of two edges in the general problem to a single edge in a reduced problem, which is possible due to the construction of G. The reduced graph $\hat{G} = (\hat{V}, \hat{E})$ can then be represented as shown in Figure 3.

In Figure 3 nine scenarios are represented as a directed graph, where each edge is a connection type, and each path

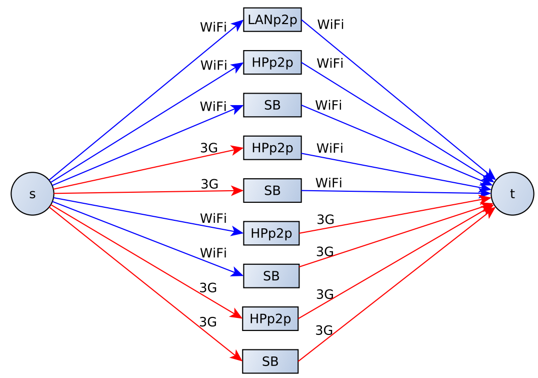

Figure 3: The reduced graph of the possible connection combinations.

from s to t is a combination of connections. The LANp2p only has WiFi to WiFi because both devices are on the same LAN, and hence both using WiFi, while the HPp2p and the SB has 4 combinations each because of the 2 connection types.

A. Minimization Step

To obtain the solution to the specific scenario, the original constraints from the general problem is used except for 6), which is restated as $\sum_{(u,v)\in\hat{E}} b(u,v) = 2$ to ensure exactly one connection. The problem then reduces to minimize:

$$\begin{aligned} &\min\{g(s,v)+g(v,t)\} \\ &\text{for } v \in \hat{V} \setminus \{s,t\} \wedge \min\{C(s,v),C(v,t)\} \geq f_{min} \end{aligned} \tag{3}$$

where $f(u,v)$ in $g(u,v)$, assuming maximizing the flow on the connection, is given by:

$$f(s,v) = f(v,t) = \min\{C(s,v),C(v,t)\} \tag{4}$$

B. Prediction step

The min-cost max-flow problem as stated earlier, determines the best strategy at time t. However, since the devices are mobile, a prediction step is needed to determine for how long the link can be sustained. In the following algorithm, n denotes the nth iteration of the min-cost max-flow problem, which is calculated on each network change. b_n denotes a vector of all states $b(u,v)$. The prediction step is outlined in Algorithm 1.

Algorithm 1 Best Link Algorithm (BLA)

```
if b_{n+1} ≠ b_n then
    if handover is profitable or Θ(b_n, t) = 0 then
        Change link to b_{n+1}
    end if
end if
```

A vertical handover is considered profitable if the inequalities (5), (6) and (7) are all fulfilled. The inequalities respectively determines whether a) the increased flow from doing a handover exceeds the lost flow during handover downtime

(5), b) whether the transfer will be able to complete with the current link within the time of a handover to the better link (6) and c) whether the state can be sustained long enough to actually do a handover (7). The handover time is determined by the specific wireless interface. The following inequalities are used for the case with linear network models, but are easily mathematically extended to non-linear cases.

$$\left(\sum_{u\in V} f_{n+1}(s,u) - \sum_{u\in V} f_n(s,u)\right)\gamma(b_{n+1},t) > ... \\ ...\lambda_1\left(H(b_n,b_{n+1}) + H(b_{n+1},b_n)\right)\sum_{u,v\in V} f_n(u,v) \quad (5)$$

$$(H(b_n,b_{n+1}) + H(b_{n+1},b_n))\sum_{u\in V} f(s,u) < ... \\ ...\left(\beta_T - \beta(t)\right)\lambda_2 \quad (6)$$

$$\gamma(b_{n+1},t) > H(b_n,b_{n+1}) + H(b_{n+1},b_n) \quad (7)$$

$\gamma(b_n,t)$ is the estimated time the state b_n can be sustained from time t. This is a geometric function, that estimates the time in which the device will stay within the coverage area of a wireless network. For simplicity, it is assumed that wireless networks has a coverage area formed as a perfect circle, and the estimation is determined using geometric and a gradient estimate of velocity from position change. $H(b_n,b_{n+1})$ is the handover time for switching between state b_n and b_{n+1}, and $\Theta(b_n,t)$ is a function which is 1 if the state is possible at time t, and 0 if the state is not possible (this is relevant if a state involving a wireless AP is no longer available, and a forced handover is required, i.e. a downgrade). β_T is the estimated total bits to be transfered of the active application, and $\beta(t)$ is the cumulated bits transfered at time t. λ_1 and λ_2 can be used to adjust the algorithm, where λ_1 controls the weighting of the handover time, and λ_2 controls the weighting of the remaining time of the transfer.

If the link is not profitable or forced, the link should be removed from $\hat{G}$ and the algorithm iterated, until no more links are available or a profitable link is found.

The prediction ($\gamma(b_n,t)$) depends on the position measurement, and if noise is present, this need to be filtered. This is done by Least Square Estimation (LSE) of the direction of the movement. The velocity is estimated by orthogonally projecting the measurement onto the LSE of the direction, and averaging over the distance of these projected points.

C. Implementation Considerations

Since the algorithm considers the entire end-to-end connection, the placement of the algorithm is critical. If the decision process were to be positioned on the clients, a layer of synchronization of network and position data would be needed, and therefore the decision process is positioned on the server. This approach also benefits the power consumption on the mobile devices, since the heavy computations therefore are not performed on the potential energy and computational constrained devices.

Since the algorithm takes a centralized placement, and requires extensive knowledge of the network and the current state of the devices, the algorithm restricts the deployment to systems, like UVN in the DAM project, which provides the required information and supporting interfaces.

V. Experimental Tests

To verify the proposed algorithm, a set of experimental tests have been performed. The tests are divided into two parts:

Simulation where the algorithm is compared to the Single Device Prediction Algorithm (SDPA) (see Appendix), to investigate the possible gain in considering end-to-end connection versus independent single device connection. BLA is also compared to the Heuristic Look-ahead algorithm (HLA) from [2], in order to evaluate complexity and the noise sensitivity compared to other close related work in the field. The optimal algorithm from [2] is not chosen due to the fact that the algorithm is required to run in real time, for which the optimal solution is too computational heavy for real time execution. To cover a wide set of scenarios, a large set of realizations of random device movements and random network maps are used.

Emulation of device movement, with BLA integrated with UVN and client components deployed on two Samsung Galaxy Tab 10.1 using real physical networks (3G and WiFi). The emulation is aimed to verify the feasibility of running the algorithm in real time, and in a real application environment. The algorithm utilizes UVN to provide extended network knowledge as described in section I.

A. Simulation and emulation details

The details of the simulation and emulation, as well as the SDPA algorithm, is left out of the main part of this paper, but is described in the Appendix.

VI. Test result

A. Simulations

In the following the aggregated simulation results are presented. We consider two cases, a) a case where perfect location determination is assumed (ideal scenario), and b) a case where location data is corrupted with noise. For the throughput evaluation an upper bound of throughput is determined, by assuming handover time to be zero and always selecting the network with highest throughput. The found upper bound is not practically feasible, but serves as a absolute upper bound for a given map and device mobility.

In Figure 4 the average throughput of the simulations are presented, in which the BLA is constantly offering a higher average throughput than SDPA as it considers the entire connection hereby avoiding some unprofitable handovers. This conclusion is supported by Figure 5, which shows

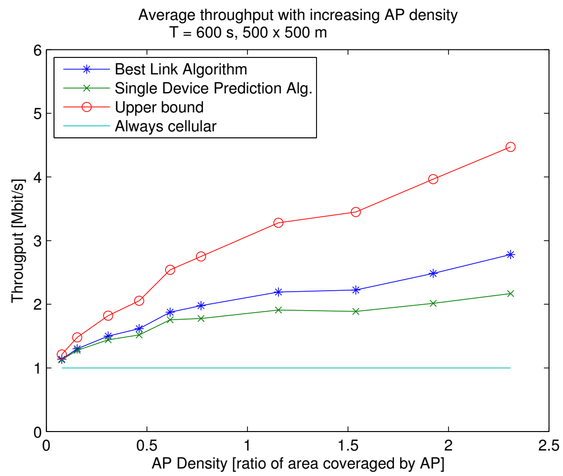

Figure 4: Average throughput offered by the respective algorithms, with incresing AP density. As seen, BLA results in a higher average throughput. The data is averaged over 1000 realizations of random walks per AP density. The upper limit is computed on the realizations in the simulations.

the average number of handovers for the BLA and the SDPA respectively, when the density of APs increases. It is apparent that the BLA triggers less handovers, while it still maintains a higher average throughput. Note that the same simulations were performed comparing BLA and HLA, which yielded similar results. However, due to the high complexity of HLA, it was not feasible to run a sufficient number of realizations to achieve statistical valid data.

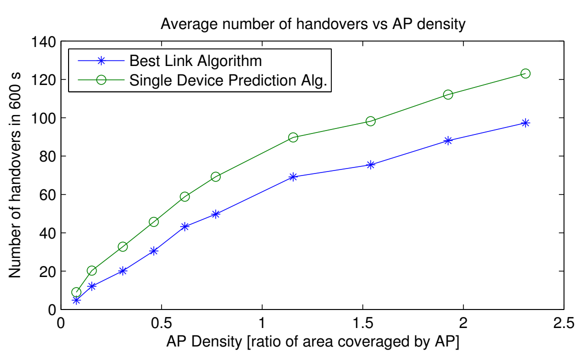

Figure 5: Comparison of average number of handovers performed compared to density of APs. Based on the same dataset as of Figure 4

The smaller number of handovers affect the duration of the connections between device and AP, which are presented in Figure 6. From this figure it is evident that the connections when using the BLA, tend to be sustained longer than with the SDPA, which leads the downtime of the connection from the devices to become lower with the BLA. This is one of the reasons the average throughput, in Figure 4, is higher for BLA than SDPA.

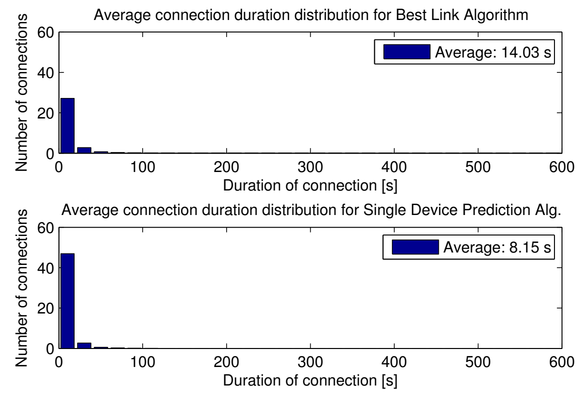

Figure 6: Duration of connections from device to device, split into 30 bins. We see a ≈72% increase in the average connection duration. Based on the same dataset as of Figure 4

Figure 7 shows that the offered throughput by both algorithms drops drastically as the handover time increases, as more and more time are spent in a offline state. But still the throughput offered by the BLA is consistently higher than the SDPA, due to the end-to-end based decision of for initiating handovers. It is also noticed, with a low handover time, the performances of the algorithms are nearly identical, which suggests a better integration of the handover time in BLA.

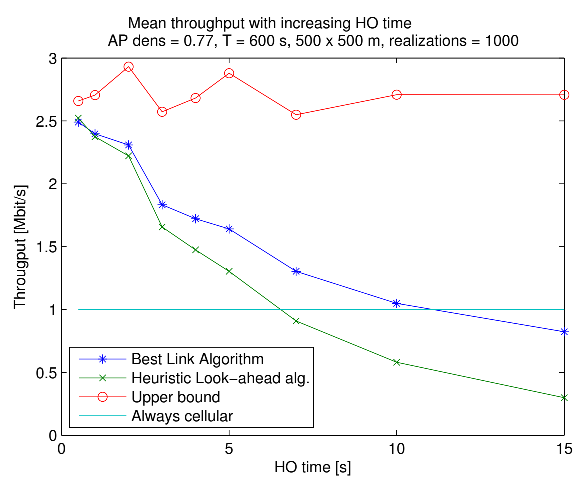

Figure 7: Average throughput offered by the respective algorithms, when handover time increases. We see that BLA is better to uphold the throughput as it incorporates the handover time more explicitly than SDPA. However, when the handover time exceeds about 10s, both algorithms fail to perform, due to the movement prediction becomes less accurate the further ahead in time is predicted.

The previous plots were based on a scenario with perfect location information. In Figure 8 a more realistic scenario is considered, as noise is added to the location data. This also causes a decrease in throughput offered by the algorithms, but with increasing handover times the BLA still offers a higher throughout than HLA. The higher drop in throughput on HLA is due to the use of a look-ahead window, in which the prediction step is very important in the decision process, whereas BLA is less sensitive to the noise.

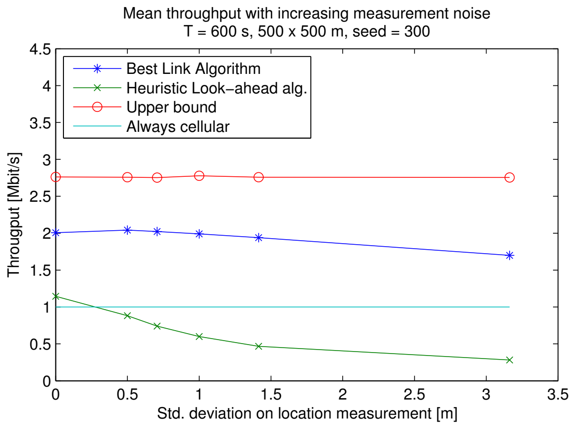

Figure 8: Average throughput offered by the respective algorithms, when adding noise to the location information. HLA deteriorates rapidly due to the look-ahead window, which makes it more sensitive to noise than BLA.

Lastly, in Figure 9 the computational time in the simulation of the three considered algorithms is shown. SDPA and BLA is more than 10 times faster than HLA (with window size of 60 seconds), which stems from not using look-ahead windowing. Note that the measurement is an estimate, and is based on the MATLAB implementation using the same implementation policies. An optimized implementation in e.g. C, will probably yield different ratios, but it is still expected that HLA is more complex than BLA and SDPA. Note that the measurement, does not consider the access delay, which is required for BLA and HLA. Factoring access delay, SDPA would be considerably lower, but the execution would also be placed on energy limited devices, whereas BLA is located on a server, and does not require heavy computations on energy limited devices.

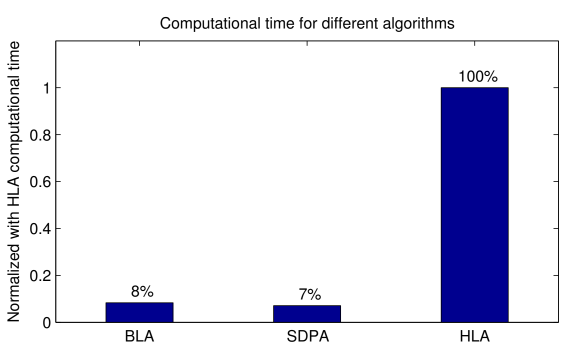

Figure 9: The computational times for the different considered algorithms, not factoring in access delay for necessary network data. BLA and SDPA is normalized with HLA, which for a single 600 s realization simulation with 5 Hz sample time took about 57s.

B. *Emulation*

The following key points are obtained by the emulations.

Figure 10 shows the results from a single test with BLA activated, with 20 MB file size and a single walk realization. As seen, when the device moves within the WiFi APs, it switches connection scheme, and improves the transfer rate. The star in the top right, is the location of 3G cell tower.

From Figure 10 it is apparent that an implementation of the BLA for real time execution is feasible as it shows network handovers and changes of transfer speed as expected. Furthermore it is seen that the BLA actually improves the transfer times of files. The transfer times for all realizations with and without BLA, illustrated using box-and-whisker diagrams (25%-75%), and as seen, the transfer time is improved when using BLA (the lower the better).

VII. Conclusion

Considering the end-to-end connection in a heuristic handover selection algorithm, we have shown, by means of simulations, that the proposed BLA results in a higher throughput with fewer handovers, indicating a better decision algorithm compared to SDPA and HLA from [2]. Also, BLA does not use a look-ahead window, which in turn lowers the computational requirement with about 90% compared to HLA. This also makes the algorithm less sensitive to measurement noise of the movement, since a predicted path in the look-ahead window is not needed.

From the emulations, it has been shown that BLA is feasible for a centralized implementation, and is able to run in real time. The emulations also showed an increased average throughput as obtained in the simulations.

VIII. Future Work

The following points could be of interest for future work:

– The BLA selects the best connection for the case of two devices. But if a device is to have connections to multiple devices simultaneously, selecting the best connection for one device may not be the best connection for another device. Therefore the algorithm should be extended to select the connection offering the best overall gain for all the endpoints.

– Define the cost $a(u,v)$ and $Q(u,v)$ from empirical measurement data to improve the correctness of the BLA.

Acknowledgment

The authors will like to thank the section of Network and Security at Aalborg University for project support and supplying two Samsung Galaxy Tab 10.1 for implementation and testing.

IX. Appendix

In this section, some of the details of the simulations and emulations will be discussed.

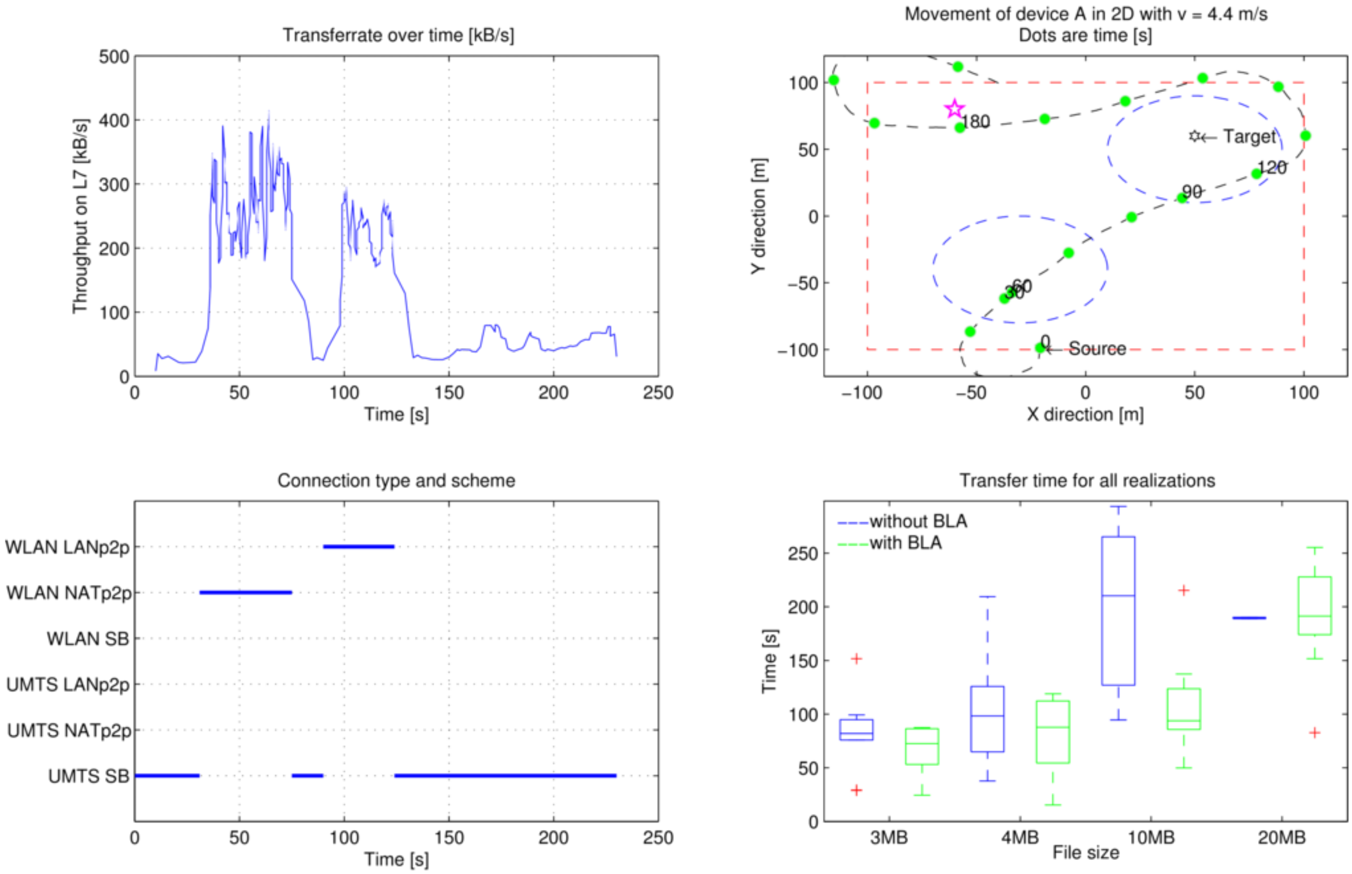

Figure 10: A single test with BLA activated, with 20 MB file size and a random walk realization. As seen, when the device moves within the WiFi coverages, it switches connection scheme, and improves the transfer rate. The transfer times for all realization with and without BLA, illustrated using box-and-whisker diagrams (25%-75%), shows that the transfer time is improved when using BLA.

A. *Single Device Prediction Algorithm (SDPA)*

The purpose of the simulation, is to investigate a possible gain in throughput and number of handovers when considering the end-to-end connection. To be able to compare with the approach of only considering a single device independent of others, a simplification of HLA [2] is constructed, without looking ahead for change in available networks. This algorithm only takes the throughput into account, and does not consider energy consumption, QoS etc. as BLA, since the gain in throughput, from these factors, are also being investigated in the experiments.

Assume $\{t_1, t_2 \;\cdots\; t_j \;\cdots\; t_J\}$ where $t_j < t_{j+1}$ is a sequence of timestamps where the network with maximum throughput $N_j^{\max}$ changes, i.e. $N_{j-1}^{\max} \neq N_j^{\max}$. Then, using a threshold to handle unprofitable handovers, at each timestamp t_j the preferred network is:

$$a_j^{\text{pref}} = \begin{cases} N_j^{\max} & \text{if } \rho - \int_{t_j}^{T} \Omega_{\max}(\tau) - \Omega_0(\tau) d\tau < 0 \\ N_0 & \text{otherwise} \end{cases},$$

where

$T = t_j + \gamma(N^{\max}, t_j)$

$\gamma(N^i, t_j)$ the time network N^i is sustainable from t_j

$\Omega_{\max}(t)$ is the throughput of $N^{\max}$ at time t

$\Omega_0(t)$ is the throughput of cellular network at time t

N_0 is the cellular network

$\rho = 2 \cdot HO \cdot \bar{\Omega}_0$

$\bar{\Omega}_0$ average throughput

HO downtime associated with handover

a_j^{pref} is the preferred network used from time stamp t_j.

B. *Mobility Model*

A mobility model has been developed, to emulate the movement of a user with a mobile device walking around an area, e.g. at campus, and staying in this area. The idea is to model the user almost walking in a straight line, but deviates a little over time. Also the model is constrained to an area of 500x500m, so when the user exits the area, he is turned back towards the allowed area by continuously turning.

The speed of the user is constant during a realization, and is uniformly distributed $V \sim U(1\frac{m}{s}, 5\frac{m}{s})$. The change in direction (d') of the device is uniformly distributed with $d'(n) \sim U((p(n) - 2) \cdot 20°, p(n) \cdot 20°)$, where $p(n)$ is the pull. $p(n) = 1$ when inside the 500x500m area with 15m padding, and $p(n) = R(n)$ when the distance to the edge is $< 15m$ from inside the area. $R(n) \sim 0.5 + Bernoulli(0.5)$ i.e. $R(n) \in \{0.5, 1.5\}$ when the distance to the edge is $\geq 15m$ and $R(n) = R(n-1)$ when distance to the edge is $< 15m$ and outside the area. When inside a WiFi AP, the walk have a 1% chance of stopping, and remain stationary for $t_s \sim U(10s, 30s)$. After a stationary period, 10s will need to pass before a new stop can occur. The position of the walk is generated at a 5Hz frequency.

C. Network Model

The network model is for simplicity defined as an affine model for the throughput, and fixed constants for the QoS parameters are set arbitrarily based on the work of [1] and [9]. It is assumed that cellular network is available always for the entire map, and that the rate is constant.

In the simulations, the WiFi throughput is uniformly distributed with 3-6 Mbit/s in upload, and 5-10 Mbit/s in download. The cellular network is fixed at 1 Mbit/s upload and 4 Mbit/s download. The network model is linear, because it is assumed that the bottleneck is the Internet connection provided by the ISP, and not the wireless technology used. The coverage radius is uniformly distributed between 20m and 50m, and the location uniformly distributed over the entire 500x500m test area. The AP density is defined as the ratio between the area covered by APs and the total area, hence due to overlap of APs, the AP density can be > 1.

For the emulation, a fixed network map of 200x200m is used, consisting of 2 APs with no overlap, given a AP density of ≈ 0.25, with bit rates as follows (download/upload) WiFi A: 20/5 Mbit/s and WiFi B: 30/5 Mbit/s. See Figure 10 for map. Real physical networks are used during emulation, and the cellular network is therefore limited by the payment plan of 4/1 Mbit/s at the carrier.

D. Test Execution

Simulation is performed with 10 network map realizations, with AP density between 0.07 and 2.30. For each network map, 1000 random walks are generated and the data is averaged over all 1000 walks for BLA and SDPA, with no measurement noise. Measurement noise simulations are performed for 1 network map, with 10 random walks for BLA and HLA, with std. deviation between 0 and $\sqrt{10} = 3.16$ m. Simulations are carried out in MATLAB. The default handover time is set to 3s.

Emulation is performed using a JAVA implementation of UVN and BLA in a distributed setting with the server/client setup as shown in Figure 1, and movement without measurement noise is emulated using MATLAB. The throughput is software limited based on position and current network. 10 random walks are generated, and for each realization 4 different file sizes are transferred from a moving transmitter, to the stationary receiver within WiFi B. Each file size and walk are tested both with BLA and using cellular network only. The tests are furthermore time limited to 5 minutes.

[10]

References

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," in *Proc. ACM IMC*, November 2009.

[2] J. J. Nielsen, T. K. Madsen, and H.-P. Schwefel, "Location assisted handover optimization for heterogeneous wireless networks," *Wireless Conference 2011 - Sustainable Wireless Technologies (European Wireless), 11th European*, pp. 1–8, april 2011.

[3] D. Xenakis, N. Passas, L. Di Gregorio, and C. Verikoukis, "A context-aware vertical handover framework towards energy-efficiency," pp. 1 –5, may 2011.

[4] B. Liu, J. Wang, and J. Li, "A joint vertical handover technique for heterogeneous wireless networks," pp. 227 –232, july 2009.

[5] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 48–49, 1950.

[6] S. R. Christensen, M. S. Dam, and L. M. Mikkelsen, "Dynamic Application Migration," September 2011. [Online]. Available: http://dam.lab.es.aau.dk

[7] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, ser. ATEC '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 1–14.

[8] D. Maier, O. Haase, J. Wäsch, and M. Waldvogel, "Nat hole punching revisited," The 36th IEEE Conference on Local Computer Networks, Tech. Rep., October 2011.

[9] R. Gass and C. Diot, "An experimental performance comparison of 3g and wi-fi," in *Proceedings of the 11th international conference on Passive and active measurement*, ser. PAM'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 71–80.

[10] F. Paternò, Ed., *Migratory Interactive Applications for Ubiquitous Environments*. London: Springer, 2011.